End-to-End Reliability Assurance of Service Chain Embedding for Network Function Virtualization

Shashank Bijwe*, Fumio Machida†, Shinya Ishida*, Seiichi Koizumi*
*System Platform Research Laboratories, NEC Corporation, Japan
†Value Co-creation Center, NEC Corporation, Japan

***Abstract*—Network Function Virtualization (NFV) enables network operators to utilize a common substrate network for offering different networking services flexibly and dynamically in the form of service chains. A service chain in NFV consists of a set of Virtualized Network Functions (VNFs) that are interconnected through a network to form a logical service. VNFs in a service chain are usually implemented on top of virtual machines hosted on commodity hardware with different reliabilities. Ensuring end-to-end reliability of a service chain is crucial for providing a highly-available service. In this paper, we propose a service chain embedding method that can assure end-to-end reliability required for service chains while improving the availability for the service and reducing the overall costs. The proposed method extends the reliability allocation method to allocate reliability goals to the VNFs in a service chain by assigning priorities to important VNFs. Through an illustrative example, we show that the present method can reduce the total downtime for the service compared to the traditional approach.**

I. Introduction

Embedding virtual networks to a substrate network is a challenging and important issue for network operators who aim to maximize the benefits from the existing substrate network by means of Network Function Virtualization (NFV). Embedding problems often involve one or more objectives to achieve while satisfying various constraints on nodes and links of virtual networks. The problem to find the optimal embedding is investigated as a combinatorial optimization problem that is typically proved to be an NP-hard problem, and hence various heuristic approaches have been proposed to address the problem [1]. A number of studies formulate the problem of optimal embedding with different optimization objectives, such as energy efficiency, resource cost, QoS, economical profit and network survivability. In the context of NFV, a virtual network is often given in the form of a service chain, in which different Virtual Network Functions (VNFs) are interlinked to form a chain fulfilling the functional and performance requirements of the requested service. For example, a Customer Premises Equipment (CPE) service may consist of VNFs for Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT), IP routing, Universal Plug and Play (UPnP), firewall and modem etc.

End-to-end reliability of a service chain is one of the important criteria to be satisfied in the embedding process. Reliability expectation for a service can be achieved by fault tolerance approaches or fault avoidance approaches. Fault tolerance approaches are usually based on redundant configuration to continue the system operation even after component failures. On the other hand, fault avoidance approaches rely on the improved component reliability to reduce the probability of failure. For reliability assurance in virtual network embedding, existing studies like [2], [4] take the fault-tolerance approach by introducing node and link redundancy for satisfying the reliability requirements for the services. For example, Chen et al. [4] proposed reliability-aware virtual network embedding to minimize the gap between realized reliability and expected reliability for the service and employ back up nodes and links to satisfy the reliability expectation of the service. Yeow et al. [2] proposed an approach to satisfy the reliability guarantees for the virtual infrastructure request by optimum allocation of redundant nodes and links. Nevertheless, Gill et al. [3] advocate that network redundancy is only 40% effective in reducing the impact of failures and comes at high cost, both monetary expenses and management overheads. In contrast, in this paper we consider a fault-avoidance approach. Considering that reliability of a VNF relies on both of the functional reliability and underlying hardware reliability, the failure probability of an important VNF could be reduced by embedding such a VNF on highly reliable resources. The importance of a VNF depends on the various factors such as the position in the service chain, the type of functionality and the operational features. A VNF with higher importance should be assigned a higher priority for being embedded on highly reliable resources.

In this paper, we propose a service chain embedding method for achieving end-to-end reliability of requested service chains in consideration with the importance of individual component VNFs. We employ a reliability allocation method and define an importance measure to make a prioritization of VNFs in the embedding process. This ensures allocation of highly reliable physical hosts for the important VNFs during the embedding phase, thus improving the end-to-end reliability of a service chain in a fault-avoidance manner.

The rest of the paper is organized as follows: In section II, we descibe the service chain embedding problem. In section III, we propose reliability allocation based service chain embedding method. In section IV, through an illustrative example we show the effectiveness of our method. Finally, section V concludes the paper.

II. Service Chain Embedding

The resource allocation process for deploying customer requests on NFV infrastructure can be divided into four steps.

978-1-5386-3285-7/17/$31.00 ©2017 IEEE

First, a customer requests a network service with functional and non-functional requirements. Second, a service provider derives a corresponding service chain for the request as per the functional requirements and computes the resource requirements to satisfy the non-functional requirements. Third, the provider determines the mapping for VNFs and virtual links of the service chain on the substrate network so that all the requirements are satisfied. Finally, following the determined mapping, the provider allocates physical hosts, launches virtual machines (VMs) to deploy the VNF software, and interconnects the VMs via virtual links to launch the requested network service. Service chain embedding is the technique used in the third step of the process and it greatly impacts on the Quality of Service and overall costs involved in offering the service.

The request for embedding a service chain is represented as a graph $G^v = (N^v, L^v)$, where N^v is a set of VNFs and L^v is a set of virtual links. Non-functional requirements can be specified by a triplet $(c(n^v),\, b(l^v),\, R(G^v))$, where $c(n^v)$, $n^v \in N^v$ is CPU demand, $b(l^v)$, $l^v \in L^v$ is bandwidth demand, $R(G^v)$ is end-to-end service reliability expectation for G^v. Meanwhile, the substrate network can be represented as a graph $G^s = (N^s, L^s)$, where N^s is a set of substrate nodes and L^s is a set of substrate links. Each substrate node $n^s \in N^s$ is associated with CPU capacity $c(n^s)$ and node reliability $R_{n^s}(t)$. Each substrate link $l^s \in L^s$ is associated with bandwidth capacity $b(l^s)$ and link reliability $R_{l^s}(t)$. Service chain embedding is represented by the VNF mapping $\sigma_N : N^v \to N^s$, and virtual link mapping $\sigma_L : L^v \to L^s$. The problem of service chain embedding can be defined as follows:

Given a service chain request $G^v = (N^v, L^v)$ with $(c(n^v),\, b(l^v),\, R(G_v))$, and substrate network $G^s = (N^s, L^s)$, find VNF mapping σ_N and link mapping σ_L, that minimize the resource cost and the reliability gap between the expected reliability and achievable reliabiliy, while satisfying the constraints on reliability, CPU and bandwidth. The objective function is formulated as follows:

$$\boldsymbol{minimize}\;\; w_r \left[\sum_{n^v \in N^v} y_N + \sum_{l^v \in L^v} y_L \right] +$$

$$w_c \left[\sum_{n^s \in N^s} \sum_{n^v | \sigma_N(n^v) = n^s} c(n^v) + \sum_{l^s \in L^s} \sum_{l^v | \sigma_L(l^v) = l^s} b(l^v) \right],$$

$Subject\ to\ Constraints\ :\ R\left[\sigma(G^v)\right] \geq R(G^v),\ 0 < c(n^v) \leq c(n^s)$ and $0 < b(l^v) \leq b(l^s)$.

Where y_N and y_L represent the gaps between the expected reliabilities and achievable reliabilities for node mapping and link mapping, respectively, w_r and w_c are weights to control the importance of reliability and the resource cost, respectively.

III. Reliability Allocation Based Service Chain Embedding

To provide a solution to the mapping problem, we divide the service chain embedding in two steps, i.e., first pre-processing and then mapping. The pre-processing step employs a reliability allocation method [5] for service chain, which interprets the expected reliability $R(G^v)$ to the reliability goals of constituent VNFs. In general, reliability allocation method involves solving the inequality:

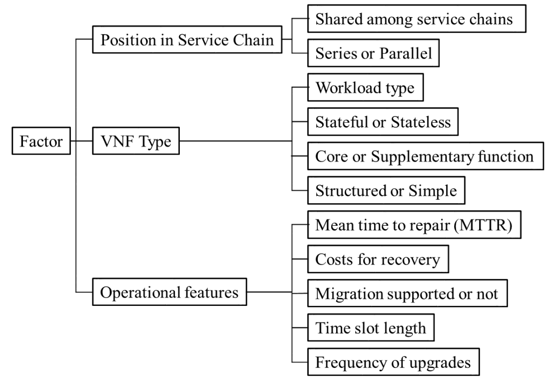

Fig. 1. Factors for Prioritizing VNFs of Service Chain

$$f(R_1, R_2, ..., R_n) \geq R_s,$$

where, R_s is the required system reliability of the system consisting of n components, $R_i, i \in [1...n]$ is the reliability allocated to the i-th component, f is the functional relationship between the component and the system. Individual VNFs in the service chain have different characteristics and they cannot be dealt with equally. Our approach takes into consideration the importances of VNFs and assigns high priority to VNFs that need high availability. In order to determine the priority of VNF in service chain, we categorize the factors to be considered as shown in Fig. 1, which illustrates the list of factors divided in three categories.

The process of determining the priority of a VNF starts with gathering the information associated with each factor from various sources of information including common VNF catalogs, centralized orchestrator, OSS/BSS or other databases in the system, etc. [6]. Then, quantitative score for each factor is computed based on the gathered information for all the factors in each category. An approach to decide the score is explained in the following subsection. Next, the scores of all the factors in the categories are added together to compute the category-wise score. Finally, the weighted sum of category-wise scores is computed as the importance measure, i.e., priority of the VNF.

A. Factors for Prioritizing VNFs in Service Chain

VNF Position in Service Chain: Depending on the VNF structural position in the service chain, a VNF failure has varying degree of impact on the overall service downtime. For example, a failure of a VNF shared among multiple service chains will affect all the associated services, in contrast to the VNF dedicated for one service chain. In addition, the failure probability of the VNF placed in series have greater degree of impact on the overall end-to-end reliability of the service, whereas failure probability of a VNF placed in parallel with similar or different type of VNF have much less impact.

VNF Type: Requirements on the reliability of VNF greatly depend on the type of workload handled by the VNF. For example, VNFs handling signal processing, switching/forwarding, HQoS, access control/ authentication typically have strict requirements for higher reliability. Assigning higher reliability to the stateful VNFs compared to the stateless VNFs is also important, since in the event of failure of stateful VNFs restoration of lost state is crucial for the service continuity, and it leads to increased downtime and increased recovery costs for the service. Another type which helps in differentiating between VNFs is, whether they execute a core function for the service or a supplementary function. A structured VNF can be more prone to failure as compared to simple VNF instances, thus ensuring higher reliability for them is also important.

Operational Features: Different VNFs have different Mean Time To Recover (MTTR) based on their implementation, size or complexity. Thus, VNFs with higher MTTR become the bottlenecks of overall service availability and ensuring higher reliability for these VNFs can improve the overall availability of the service. Also from the cost perspective, restoration of some VNFs involves more resources such as computing, networking or storage compared to others, and ensuring less probability of failure for these VNFs helps reduce the overall cost. During the operation phase, sometimes migration of VNFs is necessary either due to a failure of underlying hardware or changing service requirements. Some VNFs are not fully supported for migration and therefore may lead to service downtime during migration. Thus, ensuring higher reliability for these VNFs is also important. It is important to host these VNFs on reliable resources so that the need for migration is reduced, and to increase the overall availability of the service. Another feature is the time slot length, i.e., for a particular service chain involving multiple VNFs, each VNF processes the information for a particular portion of the time from the total processing time. Failure of the VNF having greater share of the overall processing time affects the service availability more compared to others. Upgrades of the VNF software is also essential for including new features, fixing bugs, etc., and often leads to failure of the VNF during upgrades and hence service downtime. Thus, ensuring higher reliability for the VNFs requiring frequent upgrades can reduce the adverse effect on the services.

Among the three categories, we consider VNF position has the highest priority because it has a greater impact on running services compared to other two categories. We also believe that VNF type should be prioritized over the operational features because a VNF's design and complexity play a direct impact on the probability of failure. We can incorporate such preferences over the categories by assigning weight for each category and computing the weighted sum of category-wise scores.

B. Reliability Goal Setting for VNFs of Service Chain

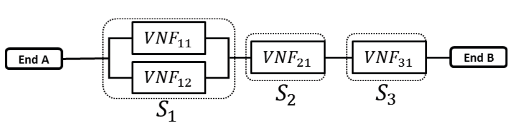

Fig. 2. Sample Service Chain

VNFs in a service chain can be connected in series, in parallel or in a combination of both depending on the service requirements. Fig. 2 shows a simple series-parallel service chain with three subsystems S_1, S_2, S_3 connected in series, where subsystem S_1 consists of two VNFs in parallel and subsystems S_2, S_3 represent individual VNFs. The normalized importance measures for constituent VNFs of a service chain is represented as p_{11}, p_{12}, p_{21}, and p_{31}. The proposed reliability goal setting steps can easily be extended to apply for more complex service chains by representing it in terms of subsystems. Here, we achieve the reliability goal setting in two steps: first, reliability goal setting for all the subsystems and then reliability goal setting for individual VNFs in the subsystem by considering the subsystem reliability goal as the target reliability. Thus, given the expected reliability for the service R, reliability goal for subsystems in series can be derived as

$$R_i = (R)^{p_i},$$

where R_i and p_i is the reliability goal and the normalized importance measure for the i-th subsystem, respectively [5]. Typically in a service chain, VNFs arranged in parallel combination to form a subsystem are of same type. Thus we compute the p_i for such subsystem as the average values of importance measures of all the constituent VNFs in the subsystem. Since in our case subsystems S_2, S_3 represent the individual VNFs itself, thus reliability goals for VNFs is same as the reliability goals for the subsystem ($R_{21} = R_2$, $R_{31} = R_3$). Next, reliability goals for VNFs connected in parallel [5] can be derived as

$$F_{ij} = (F_i)^{p_{ij}},$$

where F_{ij} is the unreliability goal of VNF_{ij} of i-th subsystem, F_i is the unreliability target for i-th subsystem. Unreliability is expressed as $F = 1 - R$.

In addition, software reliability of a VNF needs to be considered before deciding the reliability goals for the substrate network resources. Since VNF software runs on top of a hypervisor which in turn is instantiated on hardware, reliability for a realized network function is the multiplication of reliabilities of these three entities. VNF software is determined during the service chain design phase and corresponding software reliability value R^s_{ij} can be obtained from the VNF catalogs. In our approach we consider the substrate resources as the combination of hardware and hypervisor layer. Thus, the substrate reliability goal for VNFs can be derived as

$$R^{sub}_{ij} = R_{ij}/R^s_{ij}.$$

Our heuristic algorithm for service chain embedding is summarized in the algorithm 1. In algorithm 1, the lines 2 to 9 show the steps for setting the reliability goals for the VNFs in the requested service chain.

C. Embedding Phase

After the reliability goal setting phase, the service chain request associated with different substrate reliability goals for VNFs is forwarded to the embedding phase to find the

Algorithm 1 Service Chain Embedding

1: Input: $G^v, c(n^v), b(l^v), R(G^v), G^s$
// Reliability goal setting
2: **for** each VNF retrieve scores for factors
3: **for** each category add scores
4**:** multiply by category-weight & compute total score
5: **end for**
6: **end for**
7: normalize scores to obtain the values of p_{ij}
8: **for** each VNF, derive reliability goal R_{ij}^{sub}
9: **end for**
// Node and Link Mapping
10: **Minimize** reliability gap between reliability goal and substrate node reliability, where rewarding $R(\sigma_N(n^v)) \geq R(n^v)$ region and penalty for $R(\sigma_N(n^v)) < R(n^v)$ region. While satisfying overall reliability constraint and node capacity constraints
11: Compute shortest path between identified substrate nodes, which satisfies link constraints

optimal embedding. Embedding phase is further divided into two steps, node mapping phase (step 10 of Algorithm 1) and link mapping phase (step 11 of Algorithm 1).

For the sake of simplicity we consider the reliability gap objective only in node mapping phase. In link mapping phase, considering link constraints shortest path on substrate network is computed which connects the substrate nodes computed in the node mapping phase.

IV. Case Study

Due to the limited space in this short paper, we just illustrate a minimum example of a service chain embedding to show the effectiveness of our approach. Comprehensive evaluation with larger system examples will be addressed in our future work.

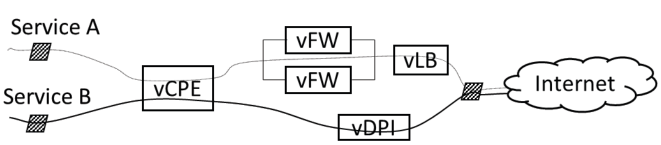

Fig. 3. Sample Service Chains

Fig. 3 shows two networking services requested by different customers, one is service A where customer requests for the advanced firewall (vFW) functionality along with virtualized Customer Premises Equipment (vCPE) and virtualized Load Balancer (vLB) functionality and the other is service B where customer demands to control traffic from specific applications using virtualized Deep Packet Inspection (vDPI) for its vCPE service. Computing requirements such as number of CPU cores and networking requirements such as bandwidth are decided during the service modelling phase. Overall reliability expectations for the service A and B are assumed to be 95% and 96%, respectively. In this example, we consider three factors for prioritizing VNFs in a service chain. First, shared, where VNFs are assigned scores as per the number of service chains the particular VNF is part of. Second, state of the VNF,

TABLE I

VNF Type	{shared,stateful,MTTR}	Total Score	p_{ij}	$R_{ij}^{sub}(t)$
vLB	{1, 0, MTTR(1min):0}	1	4/9	0.9774
vFW	{1, 0, MTTR(30sec):0}	1	4/9	0.8496
vCPE	{2, 1, MTTR(6min):1}	4	1/9	0.9943
vCPE	{2, 1, MTTR(6min):1}	4	1/3	0.9865
vDPI	{1, 1, MTTR(1min):0}	2	2/3	0.9731

i.e., score 1 is assigned if the VNF is stateful and 0 if the VNF is stateless. Third, MTTR, in this example for simplicity we use threshold to be 3 min; i.e., if MTTR is more than 3min score is 1 else 0. For simplicity, we consider above three categories to have equal importance, so we set the category-wise weight to be 1. Table 1 shows the corresponding values for scores, normalized importance measure and reliability goal for VNFs.

In a simulation, we considered substrate network with 100 nodes with varying reliability level and service chain requests with varying MTTR values, we applied proposed embedding algorithm and compared the effectiveness of our approach with the traditional reliability-aware embedding approach [4] without reliability goal setting. Our approach achieves reduced average service downtime as shown in Fig. 4. Consideration of other factors in an experimental evaluation and quantifying their effect in further reducing the downtime is a part of our future work.

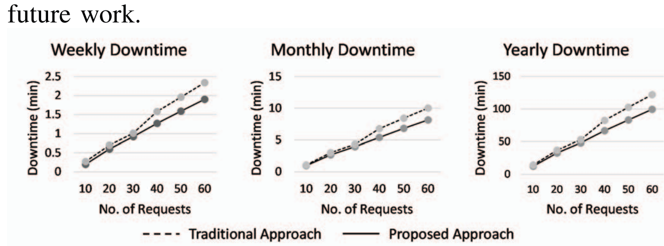

Fig. 4. Service Downtime Comparison

V. Conclusion

Our approach focuses on satisfying end-to-end reliability for a network service, by setting reliability goals for individual VNFs in accordance with their priorities in the service chain. As the more reliable hardware resources are assigned for critical VNFs of the service chain, the degree of adverse impact of failures is reduced over the lifetime of the service. An illustrative example is presented to show the effectiveness of the approach. More extensive study by simulation experiments with real industrial case examples is included in the future work.

References

[1] A. Fischer et al., "Virtual network embedding: A survey," IEEE Communications Surveys & Tutorials 15.4 (2013): 1888-1906.
[2] W. Yeow, W. Cédric and K. Ulas, “Designing and embedding reliable virtual infrastructures.” ACM SIGCOMM Computer Communication Review 41.2, 2011, 57-64.
[3] P. Gill, N. Jain, N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications." ACM SIGCOMM Computer Communication Review, Vol. 41, No. 4, ACM, 2011.
[4] Y. Chen, A. Sara, A. Chadi, “Corner: Cost-efficient and reliability-aware virtual network redesign and embedding,” Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on. IEEE, 2014.
[5] M. Catelani et al., “Reliability Allocation Procedures in Complex Redundant Systems,” IEEE Systems Journal, 2017.
[6] Network Functions Virtualisation (NFV), Reliability, Report on Models and Features for End to End Reliability, ETSI GS NFV-REL 003 V1.1.1, April 2016.